Outcome-Based Approach in Teaching Digital Systems Design for Undergraduate Computer and Electronics Engineering Programs

C. Llorente
Department of Electronics and Communications Engineering, De La Salle University, Manila, Philippines
cesar.llorente@dlsu.edu.ph

Abstract—**The adoption by the Philippine Educational system of the K to 12 program, which puts an additional 2 years in the secondary education of high school students, prompted Higher Educational Institutions particularly offering engineering programs to rewrite the curriculum. With the new curriculum for all engineering programs already at the public consultation stage, the contents of the teaching materials must be adjusted to comply with the Outcome Based Education (OBE). In this work, an outcome-based laboratory manual for teaching digital systems design has been developed, and the approaches used in its adoption are described. Key features of the manual are the tutorials section which introduces VHDL and FPGA as the hardware modeling and implementation tools, and then followed by machine problems which are small projects that can be designed and implemented a team of students up to 5 members. The machine problems after implementation can be reused to develop the term project which has a higher level of complexity. Three ways of administering the course using the laboratory manual in three consecutive academic years are presented. Based on the experience, it is surmised that approach requiring students to devise their own experiments from solving machine problems while integrating the results of the tutorial to the theoretical framework section of the experiment document provided the balance between tutorial-based learning and output-based learning.**

Index Terms— **Digital Systems Design; FPGA-Based Design; Outcome-Based Education; VHDL.**

I. Introduction

The adoption of outcome based education as a result of the realignment of tertiary education to the additional two years of basic education places emphasis on outcome based education (OBE).

The goal is to make engineering education in the country responsive to meet the demands of global equivalency in engineering programs [1]. Towards this end, one of the courses in the Department of Electronics Engineering at De La Salle University particularly the Digital Systems Design is one of the fundamental core courses in Computer and Electronics Engineering that was rewritten using the OBE format. The course provides the foundation for succeeding courses such as Computer Architecture and Embedded Systems for the Computer Engineering students. Hardware and software tools are used to facilitate teaching the concepts and to introduce software design tools and hardware prototyping tools. Teaching the course employed various methodologies and software together with hardware tools to facilitate learning.

The teaching of Digital Systems Design has been done with the use of hardware or software tools or both. This has been demonstrated in [2] where the use of software tools in the form of a compiler was written to simulate the operation of digital systems. Similarly, the use of hardware was used to verify the operation of digital design in actual hardware. A patch board attached to a 6800-based computer allows simulation of logic blocks and interconnections of more complex circuits [3]. The configurations of logic blocks in the patch board are specified by statements written using special purpose language. A sequence of statements written in special-purpose language specifies the configurations and interconnections of logic blocks in the patch board. The use of Erasable Programmable Logic Devices (EPLD's) in implementing useful functions such as an asynchronous modulo N counter, shift register and synchronous counter was presented in [4], where the erasability and re-programmability features of the EPLD are exploited in implementing logic functions and justifying high initial equipment cost.

In the early 90s, to enhance competitiveness, companies engaged in design and development of electronic equipment started to employ Computer Aided Design and Automation Tools to shorten product time to market. This raises concerns over EE undergraduate that have requisite skills set to use the CAE tools and methodologies. In response to this, laboratory oriented courses were developed covering digital systems design and design of equipment using embedded microprocessors [5]. VHDL which stands for VHSIC (Very High Speed Integrated Circuit) – HDL (Hardware Description Language), is one of software tools used to model digital circuits. In [6], VHDL was used in teaching digital systems design course which allows students to describe digital systems at a behavioral level, at a logic equation level and as an interconnection of components. The digital system can be specified at the behavioral level and then tested using a simulator. The system can be refined leading to a structural model closely resembling the actual hardware implementation.

To observe the behavior of a design in actual hardware after modeled in VHDL, reconfigurable hardware such as the Field Programmable Gate Array is used. After the design has been tested by simulation, the design can be implemented on a programmable device such as a FPGA. In [7], a reconfigurable hardware is used implement the design modeled in VHDL. Digital system design skills are strengthened through the use of an interdisciplinary problem-based model. While most approach in teaching digital systems design is to teach VHDL as a component modeling tool and how to use the EDA tools for synthesizing and simulating the design, [8] proposed an approach to teaching digital systems design by emphasizing on the functional verification of hardware design. The results of the study suggest that their approach improves on the learning process and the achievement of learning the concepts and skills in digital design. An innovative project-based approach of teaching digital system design to computer engineering undergraduate is proposed by [9], where students design and implement industry-like projects through a series of laboratory exercises. The effectiveness of the method is attested by the result where 62 percent of the senior design projects are attributed to the concepts learned from the innovative approach. Similarly, [10] integrates Information and Communication Technologies in the delivery of content to teaching digital systems design.

Building a simple microprocessor to teach digital systems design is presented in [11] and [12], wherein simple processor architecture is used as an example in building digital circuits. Multidisciplinary approached was used that allows the use of the processor architecture for computer programming in the computer science area. Using unified hardware platform to teach digital systems design and to build interface logic to control devices such as audio, video, communication and memory was presented in [13]. The platform was used in the entire computer engineering curriculum. Another approach in teaching digital systems design that aims to promote the innovative skills of computer major students early on was carried out by [14]. Selected students who participated in the open-ended experiment were given the materials at the start of the year and allowed to carry out the experiments on their own. For a period of 5 years, the study indicated that increasing number of students opted to the join the innovation experiment indicating that students are motivated to perform tasks they are interested, they have fun and success doing it and are rewarded for their efforts and success.

For a digital systems design course that caters for both computer engineering and electronics engineering programs at De La Salle University, the right mix of digital circuit design exercises must be provided to prepare students to their more advanced courses in computer architecture, embedded systems, IC design, signal processing, digital communications, and control systems courses. In this work, a laboratory manual was developed and adopted in teaching digital systems design course to computer and electronics engineering students at De La Salle University. Three ways of administering the course using the adoption of the manual are presented.

II. Laboratory Manual Organization

The laboratory manual is developed to be output-based, conforming to the OBE framework. Step-by-step tutorials facilitate learning and application of the VHDL language syntax in modeling combinational and sequential logic circuits and provide a guided tour on the use of the ISE 14.x Electronic Design Automation software and the Spartan 3E FPGA Development Board. Machine problems at the end of each tutorial section allow students to develop and exercise creativity in developing solutions creating VHDL model of digital circuits. The EDA simulation tools provide students the ability to verify functionality and performance of their design, and the FPGA development board enables students to observe the operation of their design in actual hardware.

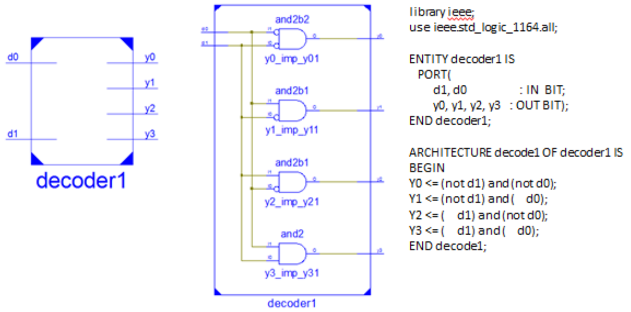
A. Tutorial Section

A step-by-step tutorial guides the students on how to use the Xilinx ISE14.x Tool suite and to the Spartan 3E FPGA Starter Kit. The laboratory activity topics are grouped according to how the topics in the introductory course in digital logic. After the walk through on how to use the EDA tool and how to download design into the FPGA board, it is followed by the combinational logic modeling, arithmetic circuits modeling sequential logic modeling and then Finite State Machines. Table 1 illustrates the topics covered under the tutorial section of the laboratory manual.

Table 1
Contents of the Tutorial Section

Week	Title of activity	Description of Activity
1	Orientation. Lab Activity 1. Familiarization with the software and hardware tools	A simple VHDL code is synthesized, simulated and downloaded to a FPGA board to demonstrate the complete FPGA Design Flow.
2	Lab Activity 2. Combinational circuits modeling	Concurrent statements of the VHDL language are used to model combinational circuits such as a 2x4 decoder and a 2-1 multiplexer.
3	Lab Activity 3. Arithmetic Circuits modeling	Components such as full adder circuit and the full subtractor circuit are modeled and re-instantiated to form parallel adder-subtractor circuit which constitutes the arithmetic circuit.
4	Lab Activity 4. Sequential circuits modeling	Modeling the sequential circuit elements called flip flops, and sequential circuits such as counter and parallel load registers.
5	Lab Activity 5. Finite State Machine modeling	Modeling of Mealy and Moore FSMs from a state chart and from an ASM chart.
6	Lab Activity 6. Simple as Possible (SAP) computer modeling	Datapath and the hardwired control unit of SAP is synthesized and simulated.
7	Lab Activity 7. Modeling control unit as FSM	Control unit is modified as a finite state machine and demonstrate the flexibility of expanding features by adding more instructions.

The tutorial section provides a detailed procedure on how to model different digital circuits in VHDL. Figure 1 shows the logic symbol and logic circuit diagram generated by the tool when the VHDL model is synthesized using the EDA tool.



```
library ieee;
use ieee.std_logic_1164.all;

ENTITY decoder1 IS
  PORT(
     d1, d0            : IN  BIT;
     y0, y1, y2, y3  : OUT BIT);
END decoder1;

ARCHITECTURE decode1 OF decoder1 IS
BEGIN
Y0 <= (not d1) and (not d0);
Y1 <= (not d1) and (   d0);
Y2 <= (   d1) and (not d0);
Y3 <= (   d1) and (   d0);
END decode1;
```

Figure 1: Modeling of a 2x4 Decoder Combinational Network in VHDL using the VHDL code in column 3. Logic diagram in column 2 and the top-level logic symbol generated by the EDA tool.

Figures 2 to 3 shows VHDL modeling of a full adder arithmetic circuit and then used as a component to model a 4-bit parallel adder circuit. This shows how to combine circuit models to model more complex circuits. The functionality of the modeled circuits is verified through the waveform timing diagrams generated by the test bench code written also in VHDL in order to test the function of the modeled circuit.

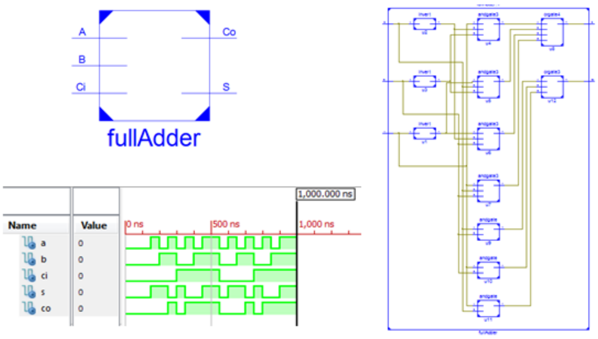


Figure 2: Modeling of the Full Adder as one of the arithmetic circuits to be modeled in Lab Activity 3

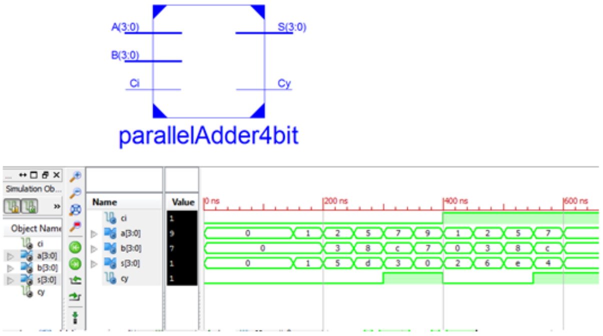


Figure 3: Modeling of a Parallel Adder as one of the arithmetic circuits to be modeled in Lab Activity 3

The binary counter is an example of a sequential network where in this activity is used as a clock divider circuit. The most significant nibble of the counter is connected to LEDs so that the counting sequence can be observed visually. The lower byte connects to a multiplexer in order to implement a variable frequency clock divider circuit, where the output is observed in an oscilloscope.

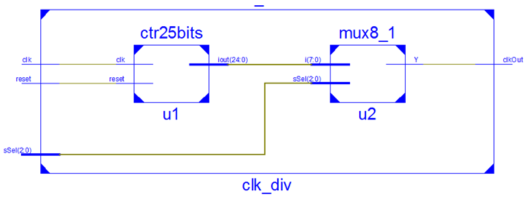


Figure 4: Modeling of Sequential Circuits in Lab Activity 4 combining combinational circuits modeled in previous lab activity

With the basic knowledge in modeling combinational and sequential circuits, modeling more complex circuit such as the datapath of a simple computer can be carried out. Figure 5 is the logic diagram of the resulting datapath of the Simple-As-Possible (SAP) computer architecture. The datapath contains data routing circuit modeled with multiplexer and decoder circuits, arithmetic logic unit modeled with arithmetic circuits, and registers and counters modeled with sequential circuits.

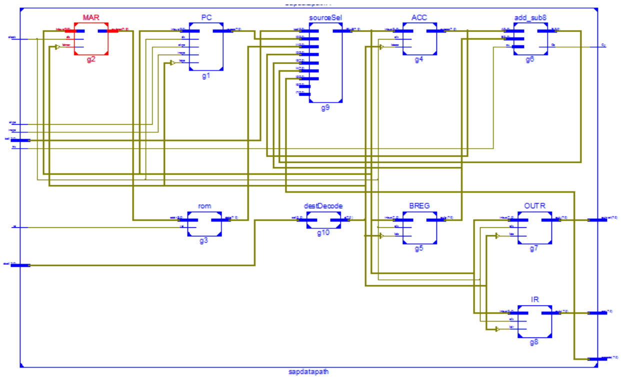


Figure 5: Modeling of the datapath of a simple computer architecture integrating combinational and sequential circuit modeling concepts learned in previous lab activities

Figure 6 shows the modeling of the Control Unit of SAP using "hardwired control". Concepts in computer instruction set classifications, and the relationship of hardware circuits needed to support the instruction is introduced. An appreciation of the relationship between hardware and software is developed.

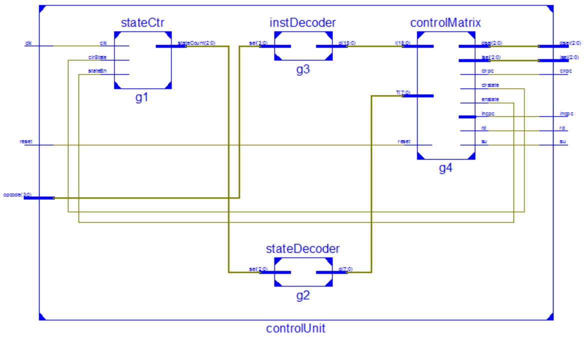


Figure 6: Modeling the control unit of the simple computer architecture

Integration of the control unit and the datapath is shown in Figure 7. The operation of the SAP architecture is demonstrated either by running a test bench through simulation or by downloading the design to the Spartan 3E FPGA Starter Kit. To see the transition of the output display as the SAP instruction is executed one at a time, a divided clock is required. This can be done by connecting the slowest clock output of the clock divider circuit in Figure 4 to drive the SAP clock input.

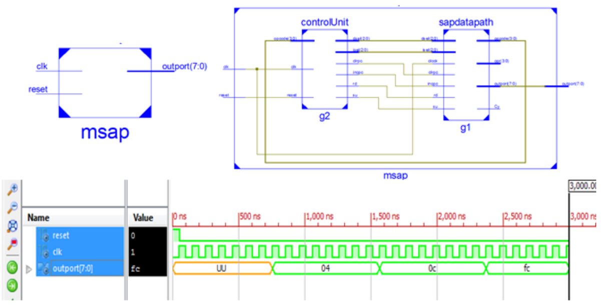
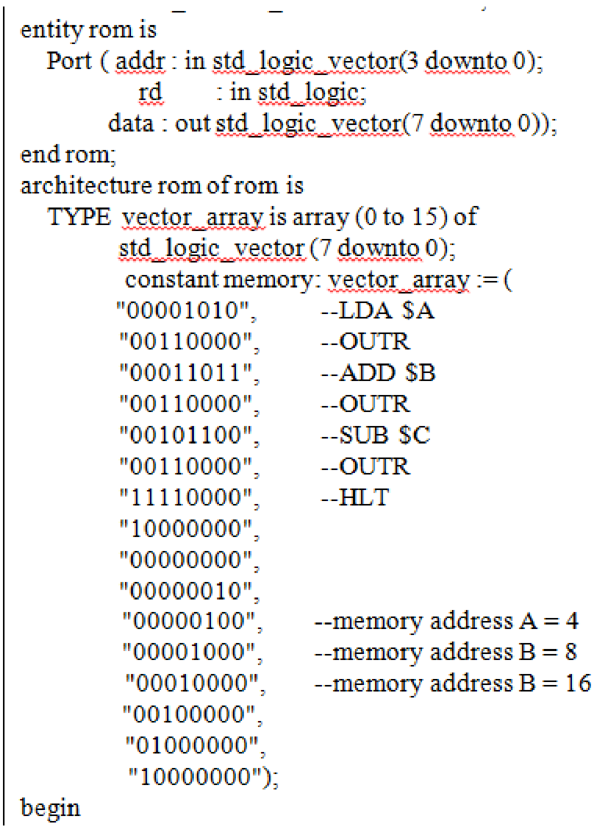


Figure 7: Integrating the datapath and control unit to model the simple computer architecture

The SAP computer program written in SAP assembly language and hand assembled to machine code is shown in the code snippet for the VHDL model of the Read Only Memory (ROM) in Figure 8. Tracing the assembly code and the timing waveform, the relationship between the program code and the output generated at the output port can be established.



```
entity rom is
   Port ( addr : in std_logic_vector(3 downto 0);
             rd        : in std_logic;
          data : out std_logic_vector(7 downto 0));
end rom;
architecture rom of rom is
   TYPE  vector_array is array (0 to 15) of
          std_logic_vector (7 downto 0);
           constant memory: vector_array := (
          "00001010",        --LDA $A
          "00110000",        --OUTR
          "00011011",        --ADD $B
          "00110000",        --OUTR
          "00101100",        --SUB $C
          "00110000",        --OUTR
          "11110000",        --HLT
          "10000000",
          "00000000",
          "00000010",
          "00000100",       --memory address A = 4
          "00001000",       --memory address B = 8
          "00010000",       --memory address B = 16
          "00100000",
          "01000000",
          "10000000");
begin
```

Figure 8: Code snippet of the VHDL model of the Program Memory of the simple computer architecture

B. Machine Problem Section

Students are expected to design, synthesize and simulate the digital components in the Machine problem section. There are designs that are downloaded into the FPGA board to observe the behavior of the design in the actual hardware. It also serves to reinforce the concepts in the tutorial section and cement the mastery of the use of the EDA software tool and the FPGA hardware development board. The list of activities and the corresponding sequence is given in Table 2.

C. Projects

For the remaining 6 weeks, students implement the system they proposed in Machine Problem 1. Additional materials in the form of VHDL code are provided that serve as templates where students can build their project upon. Students are expected to use other resources to facilitate completion of the project. Table 3 lists these additional materials.

III. Course Administration Using The Lab Manual

Three approaches were used for the introduction and adoption of the laboratory manual. The first year the lab manual was introduced, the focus is on the tutorial section of the lab manual. The step-by-step activities are carried out during the laboratory classes. In this approach, the machine problems were left only as exercises to be performed by students on their personal computers and on their own time. Solutions to the machine problems were presented as simulations only and are included in the final report.

Table 2
Contents of the Machine Problem Section

Week	Title of activity	Description of Activity
1	Machine Problem 1. Project Proposal	Prepare a proposal for a design simulation and implementation of a digital system using VHDL and FPGA, based on the expected knowledge and skills to be learned from the course, and based on the features and capability of the resources in the laboratory provided. The project can be a product, a system or a device, or an algorithm implementation.
2	Machine Problem 2.	Model and simulate the 3x8 decoder and the octal 8-1 multiplexer which will be used as part of the datapath of SAP.
3	Machine Problem 3.	Synthesize and simulate an 8-bit adder circuit using structural modeling using the 4-bit parallel adder and discuss the performance of the design if it will be used to implement a 32-bit adder and performance penalties. Support your discussion with simulation results and propagation delays data determined by the synthesis tool. Synthesize and simulate the Arithmetic Logic Unit specified in chapter 4 of the book Computer Organization and Design, 2nd edition by Patterson and Hennessy.
4	Machine Problem 4.	Implement a cascadable, bi-directional decade counter that can count up or can count down depending on the state of two control inputs, enb and dir. Also, model an 8-bit universal shift register can shift the bits to the left or to the right depending on the state of an input that controls the direction of the shift. Finally, Synthesize all registers and counters discussed in “VHDL Implementation of a Simple As Possible Computer”.
5	Machine Problem 5.	Realize a 4-bit binary multiplier using the Mealy FSM to realize the controller for the multiplier. Model the controller and the datapath separately.
6	Machine Problem 6.	Modify the datapath and the control unit of SAP to include the unconditional jump instructions. Write a program in SAP assembly that will implement an 8-bit count up binary counter. Demonstrate your design on the FPGA board. Create a clock divider circuit to slow down your clock. Modify your assembly program for a count-down counter.
7	Machine Problem 7.	Modify the control unit for SAP so that the instruction format is variable length. This means that the opcode is 8 bits (1 byte) and the operand is also one byte. Retain the same instruction set. Document your work and demonstrate your system on the Spartan 3E FPGA Development Kit.

Table 3
Additional Laboratory Activities to Assist in Project Construction.

Week	Title of activity	Description of Activity
8	Synthesizing an 8-bit RISC Datapath.	A VHDL model of an 8-bit RISC Datapath for those interested in microprocessor architecture development.
9	Controlling the ADC and DAC through the SPI Interface.	An SPI FSM to control the ADC and DAC through the SPI interface.
10	Synthesizing the VGA Compatible Display Controller.	Model of a controller to drive a VGA compatible display.

The second approach tried in the second year of implementation, was to require the students to perform the tutorials as pre-lab activities in their personal computer where they can do synthesis and simulation. Solutions to machine problems are then formulated by the students by creating the design of the circuits, formulating the procedures of the experiment and design the instruments used to capture or record their data. This also become part of the pre-laboratory activity. On the day of the laboratory session, students perform the experiment they designed up to the actual demonstration of the design using the FPGA board

In the third approach, which was tried during the third year of adoption, was to let student teams provide solutions to machine problems and integrate resulting components to come up with an integrated system that is proposed as the term project. Students perform the tutorials on their own and not integrated as part of their final report. The final report is composed of three documents namely, the project technical document and two progress reports highlighting significant milestones; first is the document detailing the solutions to the machine problems and second, a project proposal utilizing the results of the machine problems and integrated with additional components that would result in a new device or system.

IV. Results

A. *Approach 1*

Class administration and management of the class is straightforward. Students followed the procedures in the lab manual, gather results, interpret simulation diagrams and observed behavior of the design on the FPGA board. A laboratory progress report is submitted at the end of each lab session using the report templates provided. At the beginning of the following laboratory session, the final report is submitted. There are regularity and a sense of rhythm in all aspect of the conduct of the course. The lab activity final report is marked or scored based on the data presented in the progress report and on the solutions provided to the machine problem. 45% of the score is derived from the data or results in the tutorial section, another 45% is obtained from the solution to the machine problem, 5% from the way the final documentation is prepared and the remaining 5% is based on the conduct and participation of the student in the lab. All these are recorded by the instructor in the Group Evaluation Sheet attached to the Progress Report of each student group.

B. *Approach 2*

In this approach, the students prepare a pre-lab report that incorporates the results of the tutorials section to the Theoretical or Conceptual Framework section of the laboratory report. Students formulate the objectives for the particular lab activity, create the VHDL model for the components and the corresponding VHDL test bench for the system described in the machine problem, lists down the procedures for carrying out the experiment and provide instruments to capture or record the data. These can be in the form of Tables, graphs, timing waveforms or simply a blank line to record an observation. Students formulate or design their own experiment specific to the machine problem. Administration of the course using this approach requires considerable time for the instructor to check the progress report and provide feedback to students before conducting or performing the lab activity. Marking and scoring of the lab report are the same with the first approach presented previously.

C. *Approach 3*

The third approach allows students to embark on project-based approach to learning digital design. By selecting instructor-suggested topics for projects, they divide the project into smaller circuits or subsystems, following the datapath-controller model of state machines design and implementation. Sample design templates are provided to students [15 – 18] for integration to the base design. The final output of the students includes final project document that details the integration and testing of the different subsystems into one whole system, progress reports that details the design of each individual subsystems and an oral presentation of the completed project.

V. Conclusion

In this work, the development of an Outcome-Based Laboratory Manual for teaching digital systems design to undergraduate computer and electronics engineering students at De La Salle University, and the three approaches in administering the course has been presented. The tutorial approach of the lab activity provided a guided tour that facilitated learning of the VHDL language syntax and uses, as well as the use of industry-based EDA tools and hardware development platforms. The availability of the report templates provided a convenient way of preparing progress reports. However, as the solutions to the machine problems are developed outside the confines of the laboratory and on the personal time of students, the certainty that the student did the work on his own is low, diminishing the impact of the problem or outcome-based learning.

The approach in administering the laboratory course whereby the results of tutorial activity becomes part of the theoretical framework section and the students are themselves designing the experiments is the ideal approach. The approach allows to a certain extent self-paced learning and conformant to the OBE framework because students develop the skills in designing and implementing digital systems using tools used in the industry, but also test their design by developing the experiments on their own.

Project–based may be ideal to train students to become proficient in designing digital systems. However, it is very challenging both on the part of the students and instructor. Students are not so conversant with the use of VHDL language and have yet limited exposure to systems design. Making them create the design of a system may prove too difficult for the students. The time demand for the instructors handling the course for checking the documentary reports in order to verify or validate the soundness of the design and to provide timely feedback will be very high.

Acknowledgment

The development of this laboratory manual was funded by the University Research Coordinating Office (URCO-05 M U 1TAY08-3TAY08).

References

[1] CHED CMO No. 37, Series of 2012, Policies and Standards in the Establishment of An Outcome-based Education (OBE) System in Higher Education Institutions Offering Engineering Programs, Commission on Higher Education, 2010.

[2] M.MICKLE, “A Compiler Level Program for Teaching Digital Systems Design,” *IEEE Transactions on Education*, Vol. E-12, NO. 4, December 1969, pp. 274 – 279.

[3] A.D. Casaca, “Teaching digital-systems design with a microprocessor-controlled patchboard*”, IEE PROC*, vol. 130, Pt. E, No. 3, MA Y1983, pp. 75–82.

[4] C. Dawson, “The use of EPLD in Teaching Digital Systems Design,” unpublished, pp. 102-105.

[5] P. von Glahn, “CAE and case studies applied to teaching digital systems design,” *IEEE Frontiers in Education Conference*, 1991.

[6] C. H. Roth, “Teaching Digital Design Using VHDL,” *IEEE Frontiers in Education Conference*, 1994.

[7] J.Loya-Hernández, J.-L. González-Vázquez, and M.I. Garduño-Mota, “Teaching Reconfigurable Hardware Using an Interdisciplinary Problem Based Model to Strengthen Digital Systems Design Skills in Electronic Engineering Undergraduates,” *International Conference on Microelectronic Systems Education (MSE'07)*, 2007.

[8] M.E. Radu and Shannon M. Sexton, “Integrating Extensive Functional Verification in Teaching Digital Design,” *IEEE Trans. On Education*, vol. 51, no. 3, pp 385 – 393, Aug. 2008.

[9] O. B. Adamo, P. Guturu and M. R. Varanasi, “An innovative method of teaching digital system design in an undergraduate electrical and computer engineering curriculum,” *IEEETransactions in Engineering Education*, 2009.

[10] K.Jelemenská, “Integration of ICT and Unconventional Teaching Approaches into the Digital Systems Design Education towards its Efficiency Enhancement,” IEEE International Conference on Emerging eLearning Technologies and Applications, November 8-9, 2012, pp. 173-178.

[11] M.Pereira, P.Viera, A.Raabe and C.A. Zeferino, “A basic processor for teaching digital circuits and systems design with FPGA,” IEEE Publications, 2012.

[12] I. Kastelan, N. Teslic and M. Temerinac,” Generic Processor for Unified Computer Engineering Learning Platform,” *IEEE Global Engineering Education Conference (EDUCON)*, 2013.

[13] D.Baneres, “Towards an Analytical Framework to Enhance Teaching Support in Digital Systems Design Course,” Ninth International Conference on Complex, Intelligent, and Software Intensive Systems, 2015, pp.148-155.

[14] S. Li, N. Zheng, C. Quan and Y.Chen, “Cultivating Students’ Innovative Practice Ability in Hardware Experiment,” International Journal of Education (IJE), Vol. 4, No. 2/3, September 2016, pp. 1-9.

[15] C. Llorente, “VHDL Implementation of a Simple As Possible (SAP) Computer Architecture,” unpublished.

[16] C. Llorente, “Implementation of the Serial Peripheral Interface Controller for the LTC 2624 DAC chip for Interfacing to the Microblaze-based Embedded System using the Spartan3E FPGA Starter Kit,” unpublished.

[17] C. Llorente, “Implementation of the Serial Peripheral Interface Controller for the LTC 6912-1 OP AMP and the LTC 1407-1 ADC for Interfacing to the Microblaze-based Embedded System using the Spartan3E FPGA Starter Kit,” unpublished.

[18] C. Llorente, “VHDL Modeling of an 8-bit RISC-alike Datapath,” unpublished.